

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Комп'ютеризована система керування радіаторним опаленням

Виконав(ла): студент(ка) IV курсу, групи СІс-44

спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

Мікула О.О.

(прізвище та ініціали)

Керівник

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Тиш Є.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Осухівська Г.М.

(прізвище та ініціали)

Рецензент

(підпис)

Млинко Б.Б.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

« ____ » _____ 2021 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

студенту Мікула Олег Орестович
(прізвище, ім'я, по батькові)

1. Тема роботи Комп'ютеризована система керування радіаторним опаленням

Керівник роботи Осухівська Галина Михайлівна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 10 » 02 2021 року № 4/7-97

2. Термін подання студентом завершеної роботи 26.06.2021р.

3. Вихідні дані до роботи Технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

Аналіз ТЗ, огляд вимог, Аналіз можливих рішень, Розробка узагальненої структури, вибір апаратних складових, вибір програмного забезпечення, розробка алгоритмів роботи, тестування.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

КС КРБ 123.175.00.01 Е1

КС КРБ 123.175.00.02 Е2

КС КРБ 123.175.00.03 Е3

КС КРБ 123.175.00.04 СА

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4. Безпека життєдіяльності та основи охорони праці	Пилипець М.І, доктор технічних наук, професор кафедри МТ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення з завданням до кваліфікаційної роботи		Виконав
2	Ознайомлення з вимогами		Виконав
3	Опрацювання літератури щодо ESP8266		Виконав
4	Підбір апаратних складових		Виконав
5	Підбір бази для написання програмної складової		Виконав
6	Робота над розділом «Аналіз технічного завдання»		Виконав
7	Робота над розділом «Проектна частина»		Виконав
8	Виконання макетування		Виконав
9	Робота над розділом «Практична частина»		Виконав
10	Виконання завдань до розділу «Безпека життєдіяльності, основи охорони праці»		Виконав
11	Оформлення пояснювальної записки		Виконав
12	Нормоконтроль		Виконав
13	Перевірка на плагіат		Виконав
14	Попередній захист КС КРБ		Виконав
15	Захист КС КРБ		Виконав

Студент

_____ (підпис)

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

_____ (прізвище та ініціали)

Анотація

Комп'ютеризована система керування радіаторним опаленням на основі ESP8266 // Кваліфікаційна робота бакалавр на здобуття освітнього ступеня бакалавр // Мікула Олег Орестович // ТНТУ, спеціальність 123 «Комп'ютерна інженерія» // Тернопіль, 2021 // с. - 50, рис. - 32, аркушів А1 – 4, бібліогр. - 17

Ключові слова: ARDUINO, ESP8266, СИСТЕМА, КЕРУВАННЯ, КОМП'ЮТЕРИЗОВАНА СИСТЕМА КЕРУВАННЯ РАДІАТОРНИМ ОПЕЛЕННЯМ, ТЕМПЕРАТУРА, РОЗУМНИЙ БУДИНОК.

Об'єктом дослідження для подальшої розробки кваліфікаційної роботи проекту стала система “розумний” будинок.

Мета проекту – розробка автоматизованої системи керування водяними радіаторами. Хоча такі проекти вже існують, але усунувши деякі недоліки системи приверне до себе більше уваги.

Завданням роботи є розробка і створення автоматизованої системи керування водяними радіаторами.

Результат – комп'ютеризована система, якою можна зручно налаштовувати температуру у приміщенні.

Для реалізації всієї логіки системи та панелі керування було обрано одноплатний комп'ютер Raspberry Pi 3. Логіка системи реалізована за допомогою середовища розробки PyCharm та мови програмування Python.

Модулі та сенсори системи реалізовано на мікроконтролерах ESP, які програмуються за допомогою середовища розробки Arduino IDE. Програмування відбувається за допомогою мови програмування C/C++.

Abstract

Computerized radiator heating control system based on ESP8266 // Bachelor's qualification work for a bachelor's degree // Mikula Oleh Orestovych // TNTU, specialty 123 "Computer Engineering" // Ternopil, 2021 // p. - 50, fig. - 32, sheets A1 - 4, bibliography - 17

Keywords: ARDUINO, ESP8266, SYSTEM, CONTROL, COMPUTERIZED CONTROL SYSTEM OF RADIATOR DRAINAGE, TEMPERATURE, SMART HOUSE.

The object of research for further development of the qualification work of the project was the system "smart" house.

The purpose of the project is to develop an automated control system for water radiators. Although such projects already exist, but eliminating some shortcomings of the system will attract more attention.

The task of the work is to develop and create an automated control system for water radiators.

The result is a computerized system that can easily adjust the room temperature.

A Raspberry Pi 3 single-board computer was chosen to implement all the system logic and control panel. The system logic was implemented using the PyCharm development environment and the Python programming language.

The modules and sensors of the system are implemented on ESP microcontrollers, which are programmed using the Arduino IDE development environment. Programming is done using the C / C ++ programming language.

/

ЗМІСТ

СПИСОК СКОРОЧЕНЬ	7
ВСТУП.....	8
1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	9
1.1 Аналіз вимог до технічного завдання	9
1.2 Аналіз можливих рішень поставленого завдання	10
2 ПРОЕКТНА ЧАСТИНА	12
2.1 Розробка узагальненої структури комп'ютеризованої системи керування радіаторним опаленням.....	12
2.2 Обґрунтування вибору апаратного забезпечення комп'ютеризованої системи керування радіаторним опаленням	13
2.3 Обґрунтування вибору програмного забезпечення комп'ютеризованої системи керування радіаторним опаленням	15
2.4 Проектування комп'ютеризованої системи керування радіаторним опаленням	16
2.5 Алгоритми роботи комп'ютеризованої системи керування радіаторним опаленням	19
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	23
3.1 Реалізація або моделювання проектних рішень	23
3.2 Тестування.....	34
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	41

					КС КРБ 123.175.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>					КОМП'ЮТЕРИЗОВАНА СИСТЕМА КЕРУВАННЯ РАДАТОРНИМ ОПАЛЕННЯМ	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>								
<i>Реценз.</i>						ТНТУ, каф. КС, гр. СІс-44		
<i>Н. контр.</i>								
<i>Затверд.</i>								

4.1	Долікарська допомога при опіках	41
4.2	Розробка заходів щодо зниження рівня шуму на дільниці цеху	43
4.3	Трудова діяльність користувачів комп'ютерних технологій.	45
ВИСНОВКИ		47
БІБЛІОГРАФІЯ		48
ДОДАТКИ		50
Додаток А. Технічне завдання.....		50
Додаток Б. Лістинги реалізації компютерної системи керування радіаторним опаленням		56

					КС КРБ 123.175.00.00 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК СКОРОЧЕНЬ

IoT – Internet of Things

IDE - Integrated Development Environment

IP – Internet Protocol

TCP – Transmission Control Protocol.

ІС – інформаційна система

КС КРО – Комп'ютеризована система керування радіаторним опаленням

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Технології в теперішній час розвиваються колосальними темпами і все частіше знаходять нове застосування в побуті. Це стосується багатьох аспектів життя людей.

Сьогодні більшість людей мають індивідуальне опалення, що дозволяє ефективніше використовувати енергоресурси для обігріву приміщення. Тим не менше, щоб регулювати температуру в окремих приміщеннях потрібно в ручному режимі регулювати температуру кожного радіатора окремо. Якщо таких радіаторів незначна кількість, то такий контроль за ними не є проблемним. Однак, практика показує, що в приватних будинках, а також в будівлях комерційного використання (офіси, склади тощо) кількість радіаторів значно більша, а, отже, і контроль їхнього стану в ручному режимі ускладнюється. Тому розробка комп'ютеризованої системи керування радіаторним опаленням (КС КРО) є актуальним завданням.

Поява на ринку різноманітних автоматизованих систем значно спростила облік та контроль за об'єктами, а поява безпроводних технологій дозволила здійснювати цей контроль в реальному часі та на великих відстанях. Не винятком стала поява систем контролю температури. Такі системи дозволяють відстежувати температурні (і не тільки) показники та відображати їх на централізованих панелях керування, мобільних пристроях тощо [11].

Поєднання систем контролю з системами керування сприяло появі спеціалізованих комп'ютеризованих систем. Такі системи замінюють ручний контроль та регулювання роботи багатьох пристроїв, зокрема і тих, які забезпечують обігрів приміщень. Можливості сучасних централізованих систем контролю температури дозволяють оптимальніше використовувати енергоресурси. Для прикладу можна налаштувати систему на зниження температури в нічний час, або задати необхідні температурні показники згідно календарного графіку на тривалий період (наприклад, коли всі їдуть у відпустку).

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

/

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз вимог до технічного завдання

Завданням кваліфікаційної роботи бакалавр є розробка власної комп'ютеризованої системи контролю температури у приміщеннях, а саме – здійснювати керування радіаторним опаленням. Система КС КРО повинна виконувати такі функції:

- керування водяними радіаторами;
- перегляд поточної температури;
- встановлення необхідного рівня температури;
- автоматизація контролю температури.

Загалом, для забезпечення ефективної роботи, система повинна мати дружній інтерфейс користувача, зрозуміле призначення функцій та наочний результат обробки інформації, належним чином функціонувати в існуючому програмному просторі в умовах експлуатації. Система повинна бути відкритою для подальшого нарощування функціональних можливостей окремих програмних модулів. Необхідним і обов'язковим є наявність простого інтуїтивно зрозумілого інтерфейсу без спеціального налаштування та навчання. Система в результаті має автоматично керувати температурою у приміщенні. Якщо температура в приміщенні відрізняється від встановленої користувачем більше ніж на $0,5^{\circ}\text{C}$, тоді радіатор відключається; якщо ж температура опускається більше ніж на $0,5^{\circ}\text{C}$ – радіатор включається. Таким чином підтримується встановлена температура в приміщенні.

					КС КРБ 123.175.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>					АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>							9	50
<i>Реценз.</i>						ТНТУ, каф. КС, гр. СІс-44		
<i>Н. контр.</i>								
<i>Затверд.</i>								

Цей модуль повинен кріпитись на радіатор до вентиля та працювати в межах температурного діапазону від 0 до +50⁰С. Також, в модуля радіатора повинен бути власний датчик температури для контролю безпосередньо температури радіатора.

КС КРО повинна функціонувати відповідно до схеми, яка зображена на рис. 1.1.

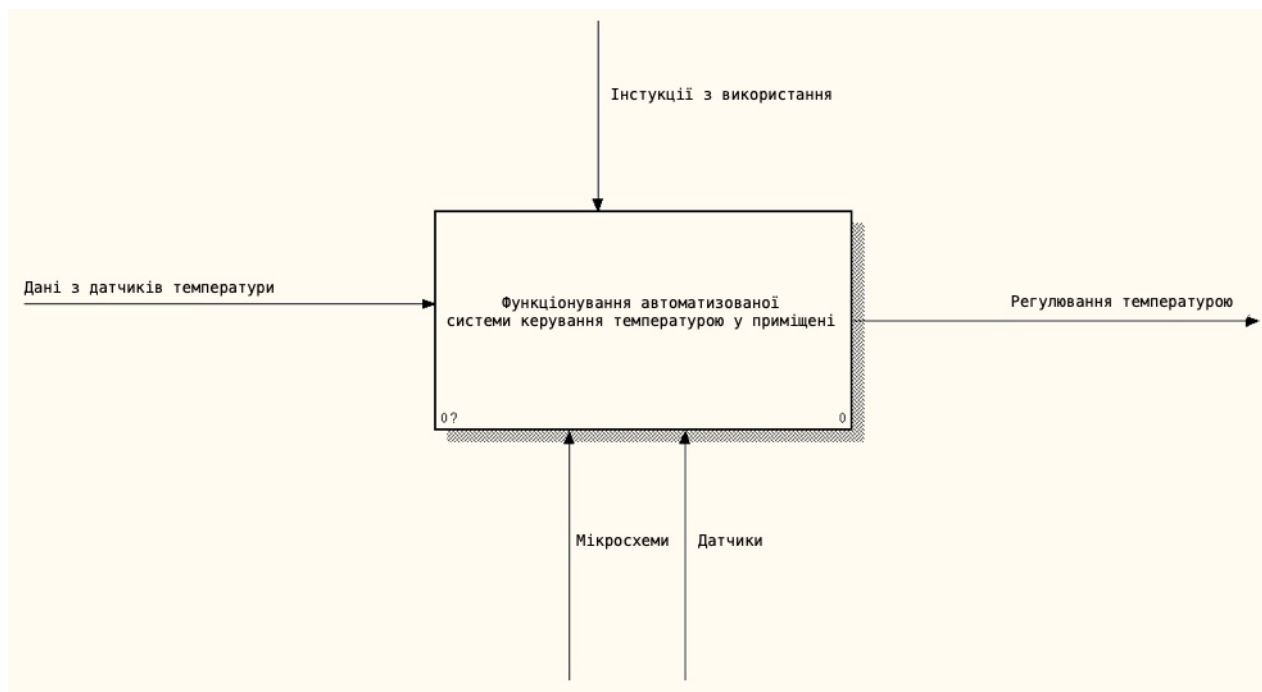


Рисунок 1.1 – Схема функціонування системи

1.2 Аналіз можливих рішень поставленого завдання

Розглянемо деякі рішення лідерів даного ринку – компаній «Danfoss» і «DEVI», які є основними виробниками обладнання для опалення приміщень, яке називається «Danfoss Link» та «DEVI snk». Ці системи працюють по технології WI-FI. Будучи в межах покриття мережі можна контролювати температуру будинку за допомогою мобільного додатку [11].

Модуль керування системою це – центральна сенсорна панель, через яку можна проводити налагодження кожного окремого радіатора по температурі . Керування радіаторами проводиться по радіозв’язку і тому проводити окремі кабелі не потрібно.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

Наступний пристрій цієї системи - це терморегулятор з датчиком температури. Призначений для визначення температури. Виконує функції включення або виключення нагріву радіатора в залежності від температури радіатора [11].

Також в системі присутні бездротові радіаторні регулятори для регулювання клапанів на водяних радіаторах.

Щоб знати поточне значення температури в приміщеннях є датчики температури повітря, які підключаються по мережі та працюють від батарейок. Цей датчик передає поточну температуру на мобільний додаток або сенсорну панель.

Наступним прикладом який ми розглянемо буде система «HTLab» від компанії «simVolt» [12]. Це система для контролю температури та вологості в приміщеннях. Сенсори автоматично проводять моніторинг температури в приміщеннях. Та, якщо температура перевищує або нижча ніж вказано в системі, сигналізує про це. Кількість датчиків, які потрібно для коректної роботи системи залежить від площі приміщення. Дана система дозволяє зменшити енергозатрати.

Незважаючи на значні переваги використання розглянутих систем, вони мають свої недоліки. Зокрема, основним стримуючим фактором для їх впровадження є ціна: вартість мінімальної комплектації становить понад 300\$. Також ця система не передбачає інтеграцію з іншими системами типу «розумний» дім та немає мобільного додатку для керування системою.

Беручи до уваги зазначені фактори, було прийнято рішення реалізувати власну комп'ютеризовану систему контролю радіаторним опаленням.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

2 ПРОЕКТНА ЧАСТИНА

2.1 Розробка узагальненої структури комп'ютеризованої системи керування радіаторним опаленням

Проектування – це комплекс робіт який складається з пошуку, досліджень та розрахунків з метою отримання опису достатнього для створення нового об'єкту або виробу, його реконструкції, модернізації, що відповідає заданим вимогам [13].

КС КРО повинна складатись з наступних модулів:

- температурні сенсори – здійснюватимуть вимірювання температури навколишнього середовища та передаватимуть ці значення по радіоканалу;
- модулі керування та вимірювання температури водяних радіаторів – здійснюватимуть контроль за проходженням потоку води через радіатор та контроль безпосередньо температури радіатора;
- головна панель керування, оснащена сенсорним дисплеєм – реалізовуватиме бізнес-логіку роботи системи та надаватиме інструменти для її налаштування;
- WI-FI роутер – забезпечуватиме зв'язок між усіма елементами системи;
- смартфон (планшет) - здійснюватиме відображення стану системи та надаватиме інструменти для керування цим станом;

Структура КС КРО наведена на рис. 2.1.

					КС КРБ 123.175.00.00 ПЗ		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розроб.</i>					<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Перевір.</i>						12	11
<i>Реценз.</i>					ТНТУ, каф. КС, гр. СІс-44		
<i>Н. контр.</i>							
<i>Затверд.</i>							
ПРОЕКТНА ЧАСТИНА							

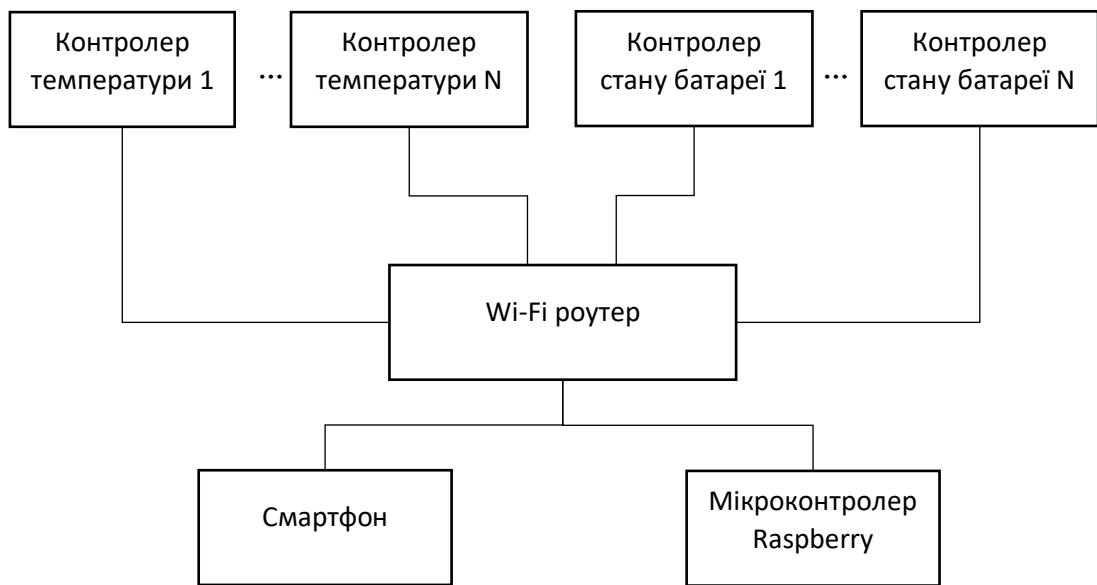


Рисунок 2.1 – Структурна схема системи

2.2 Обґрунтування вибору апаратного забезпечення комп'ютеризованої системи керування радіаторним опаленням

Для початку роботи потрібно обрати всі необхідні компоненти для реалізації системи.

Для отримання та обробки інформації з інших модулів КС КРО обрано одноплатний комп'ютер Raspberry Pi 3 (рис. 2.2).

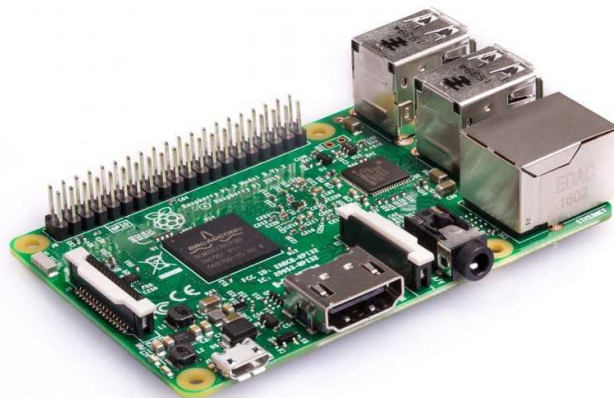


Рисунок 2.2 – Одноплатний комп'ютер Raspberry Pi 3

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Для комфортного регулювання температури в приміщені КС КРО, в додаток до Raspberry Pi 3, обрано сенсорний модуль з діагоналлю екрану 3,5 дюйми

Для реалізації модуля для надсилання даних про температуру обрано WI-FI модуль ESP8266 версії 01. Це мініатюрний WI-FI модуль на базі мікросхеми ESP8266 з вбудованим стеком протоколу TCP/IP і управлінням AT-командами. Чіп створений для використання в розумних розетках, mesh-мережах, IP-камерах, бездротових сенсорах, переносній електроніці і так далі. Одним словом, ESP8266 з'явився на світ, щоб стати «мозком» майбутнього «Інтернету речей» [7].

Для цього модуля також обрано модуль температури та вологості DHT11. Даний модуль дозволяє організувати невелику метеостанцію, як передавальний пристрій - використовується контролер ESP8266-01, а покази зчитуються з датчика DHT11. Він зібраний на платі з текстоліту має невеликі габарити 20x25мм, більшу частину місця займає датчик DHT11, поруч розташовані роз'єм для підключення ESP-01 (S) і роз'єм живлення. Для стабільної роботи ESP-01 необхідна напруга 3.3В, для цього на платі встановлений стабілізатор напруги L1336 [7].

Для реалізації модуля керування водяними радіаторами в КС КРО обрано WI-FI модуль ESP8266 версії 12. Цей модуль, як і попередній, реалізований на базі мікросхеми ESP8266, однак має значно більше портів введення-виведення, що дозволяє підключати до нього кілька зовнішніх пристроїв одночасно. Також, для виміру безпосередньо температури радіатора було обрано датчик температури DS18B20. Перевага цього датчика є те, що він є дуже точним в порівнянні з своїми аналогами і може вимірювати температуру в діапазоні від -55 до +125 градусів. Зокрема, для керування електродвигуном механічного вентиля використано два модулі електромагнітного реле.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Електромагнітне реле – це електричний пристрій, яке механічним шляхом замикає або розмикає ланцюг навантаження за допомогою магніту. Складається з електромагніту, рухомого якоря і перемикача [4].

Механічна структура системи реалізована за допомогою електричного вентиля Yongkang. Електричний вентиль в системі використовується для керування температурою водяних радіаторів. Такий вентиль працює по принципу реверсивного двигуна. Якщо подати на нього живлення, то він буде крутитись в одному напрямку, але якщо на двигуні (вентилі) поміняти полярність і подати живлення - то він буде крутитись в іншому напрямку. Електричний вентиль зображено на рис. 2.3 [9].



Рисунок 2.3 - Електричний вентиль

2.3 Обґрунтування вибору програмного забезпечення комп'ютеризованої системи керування радіаторним опаленням

Для реалізації модулів КС КРО на базі мікроконтролерів ESP8266 було обрано середовище розробки Arduino IDE – інтегроване середовище розробки (IDE) є крос-платформним додатком (для ОС MS Windows, MacOS, Linux), який написаний на мові програмування Java. Він використовується для написання і завантаження програм на Arduino-сумісні плати, а також, за допомогою плагінів сторонніх виробників на Arduino-сумісні плати інших виробників. Вихідний код

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

для IDE випускається під загальнодоступною ліцензією GNU версії 2. Arduino IDE підтримує мови C і C ++ з використанням спеціальних правил структурування коду. Arduino IDE надає бібліотеку програмного забезпечення з проекту Wiring, яка має безліч загальних процедур введення і виведення [14].

Arduino IDE використовує програму avrdude для перетворення коду в текстовий файл в шістнадцятковому коді, який завантажується в плату Arduino за допомогою програми-завантажувача в прошивці плати.

Для написаного коду потрібні тільки дві основні функції – для запуску скетча і основного циклу програми, які компілюються і зв'язуються з заглушкою програми main() в виконувану циклічну програму за допомогою ланцюжка інструментів GNU, також включеною в дистрибутив IDE [14]. Код реалізації модулів зображено у додатку Б.

Мобільний додаток на систему iOS реалізовано за допомогою мови програмування Swift та середовища розробки XCode. Swift – багатопарадигмова компільована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C і бути стійкішою до помилкового коду. Вона також пропонує безліч сучасних методів програмування, таких як замикання, узагальнене програмування, лямбда-вирази, кортежі і словникові типи, швидкі операції над колекціями, елементи функційного програмування. Основним застосуванням Swift є розробка користувацьких застосунків для macOS, iOS, tvOS, watchOS з використанням тулкіта Cocoa і Cocoa Touch. При цьому Swift надає об'єктну модель, сумісну з Objective-C [5].

Xcode — інтегроване середовище розробки (IDE) виробництва Apple. Xcode включає в себе більшу частину документації розробника від Apple та Interface Builder — застосунок, який використовується для створення графічних інтерфейсів.

2.4 Проектування комп'ютеризованої системи керування радіаторним опаленням

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

В ході виконання кваліфікаційної роботи було розроблено алгоритм функціонування КС КРО у приміщеннях.

Вхідними даними для роботи системи є:

- дані з сенсора температури в приміщенні;
- дані з сенсора температури на радіаторі;
- дані з панелі контролю системою;
- дані з мобільного додатку.

Вихідними даними є результат роботи модуля керування водяними радіаторами в процесі роботи системи.

Результатом роботи є готовий модуль, який в автономному режимі віддалено здійснює контроль температури приміщення. Якщо температура в приміщенні відрізняється від встановленої користувачем більше ніж на 0,5°C, тоді радіатор відключається; якщо ж температура опускається більше ніж на 0,5°C – радіатор включається. Таким чином підтримується встановлена температура в приміщенні.

Функціональна схема роботи системи зображена на рис. 2.4.

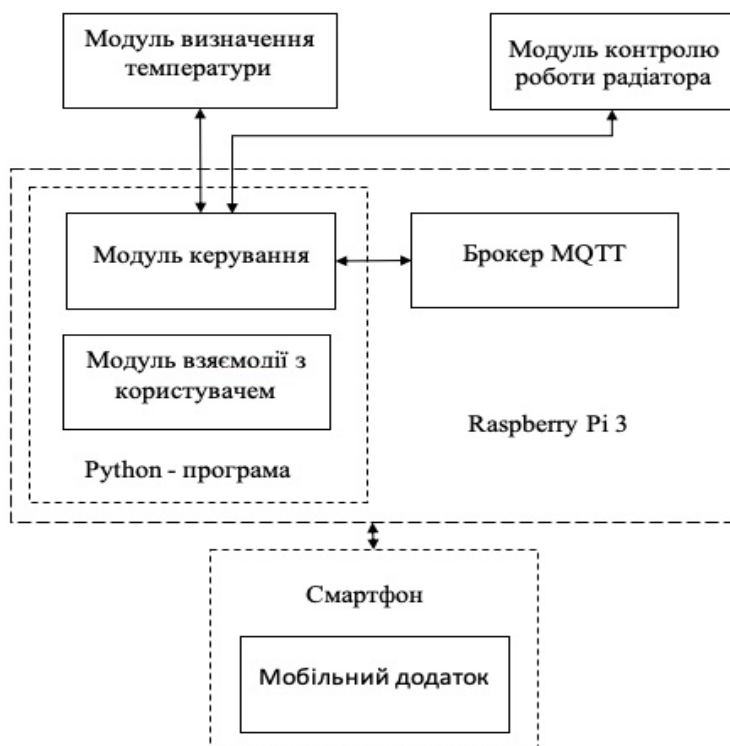


Рисунок 2.4 – Функціональна схема КС КРО

2.5 Алгоритми роботи комп'ютеризованої системи керування радіаторним опаленням

Алгоритм роботи функціонування КС КРО наведений на рис. 2.6.

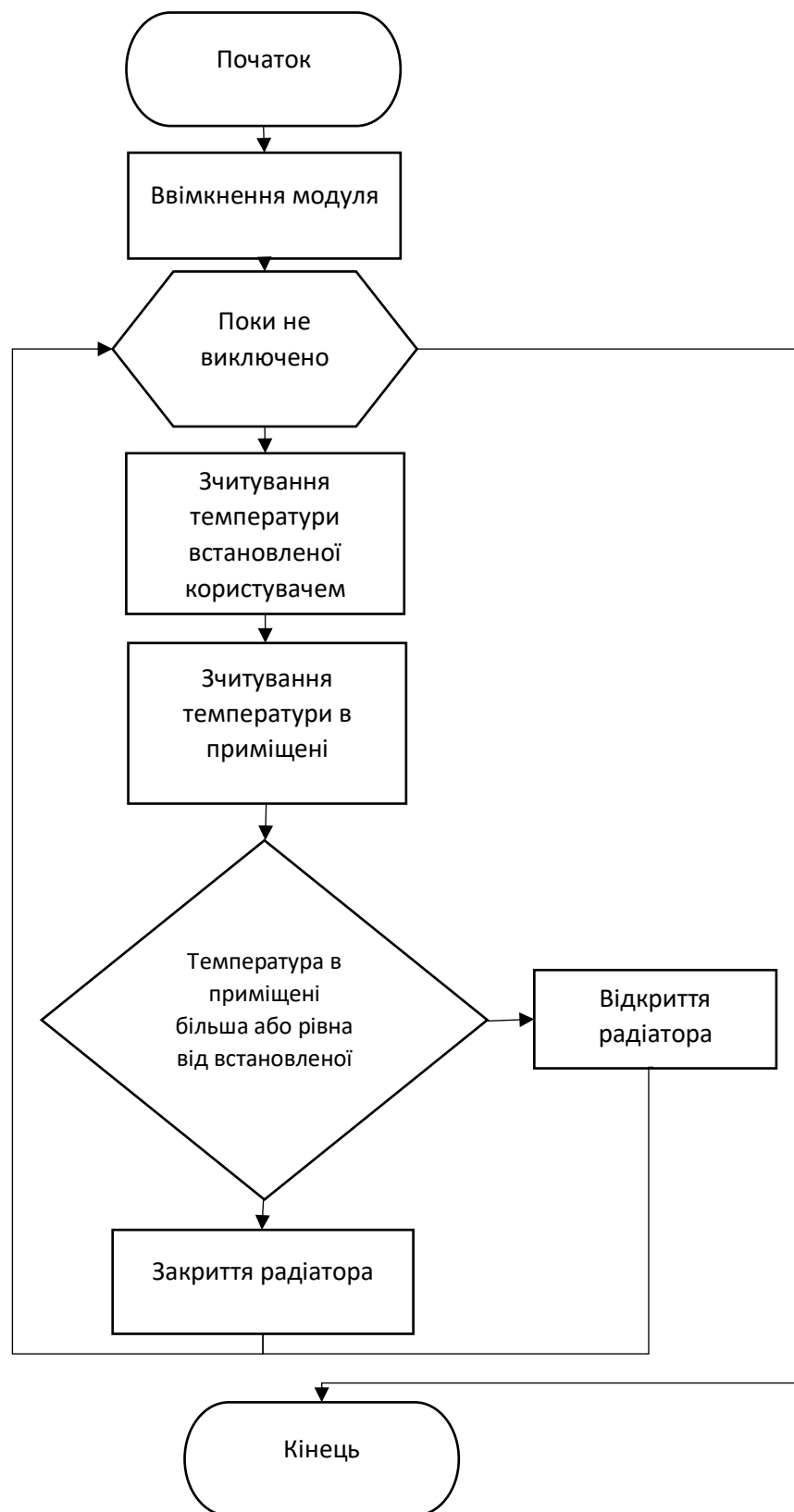


Рисунок 2.6 – Блок-схема алгоритму роботи КС КРО

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Суть алгоритму роботи КС КРО полягає в наступних кроках. Для початку відбувається ввімкнення модулів. Після чого відбувається зчитування температури, яку встановив користувач. Далше відбувається порівняння поточної температури в приміщені з температурою, яку встановив користувач. Алгоритм модуля температури зображено на рис. 2.7.



Рисунок 2.7 – Блок-схема алгоритму роботи температурного модуля

Після чого, температурний модуль відсилає сигнал на брокера і той вже після того, як отримає данні вирішує, закривати чи відкривати модуль радіатора.

Також, якщо температура залишається не змінною то брокер не відправляє сигнал дії на модуль радіатора. Алгоритм роботи модуля радіатора зображено на рис. 2.8.

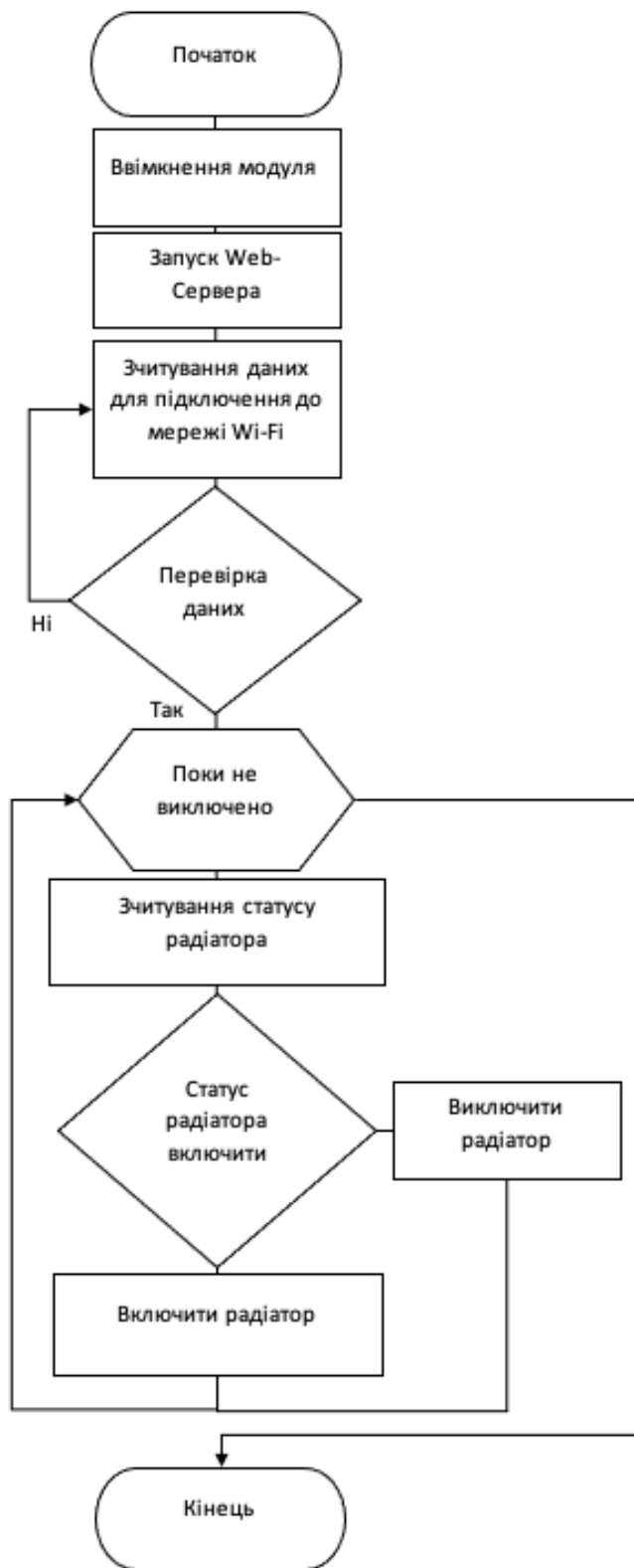


Рисунок 2.8 – Блок-схема алгоритму роботи модуля радіатор

Змн.	Арк.	№ докум.	Підпис	Дата

Таким чином, після описання алгоритмів роботи кожного модуля у вигляді блок-схема стає більш зрозуміло принцип роботи кожного з модулів КС КРО. А саме загальний алгоритм роботи КС КРО, алгоритм роботи модуля радіатора та алгоритм роботи модуля температури. Розглянувши більш детальніше кожен з алгоритмів можна побачити, що принцип роботи модулів є однаковий з головною відмінністю тільки в інформації, яку вони надсилають або опрацьовують.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація або моделювання проектних рішень

Для початку реалізації самої системи потрібно встановити на Raspberry Pi MQTT Broker Mosquitto[3]. MQTT – спрощений мережевий протокол, що працює на TCP/IP. Використовується для обміну повідомленнями між пристроями за принципом видавець-підписник. Обмін повідомленнями в протоколі MQTT здійснюється між клієнтом (client), який може бути видавцем або підписником (publisher/subscriber) повідомлень, і брокером (broker) повідомлень (наприклад, Mosquitto MQTT). Видавець відправляє дані на MQTT брокер, вказуючи в повідомленні певну тему – топік (topic). Підписники можуть отримувати різні дані від безлічі видавців залежно від підписки на відповідні топіки.

Пристрої MQTT використовують певні типи повідомлень для взаємодії з брокером, нижче представлені основні:

- Connect - встановити з'єднання з брокером.
- Disconnect - розірвати з'єднання з брокером.
- Publish - опублікувати дані в топік на брокера.
- Subscribe - підписатися на топік на брокера.
- Unsubscribe - відписатися від топіка.

Приклад роботи MQTT брокера зображено на рис. 3.1.

					КС КРБ 123.175.00.00 ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>					РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	<i>Літ.</i>	<i>Арк.</i>	<i>Акрушів</i>
<i>Перевір.</i>							23	18
<i>Реценз.</i>						ТНТУ, каф. КС, гр. СІс-44		
<i>Н. контр.</i>								
<i>Затверд.</i>								

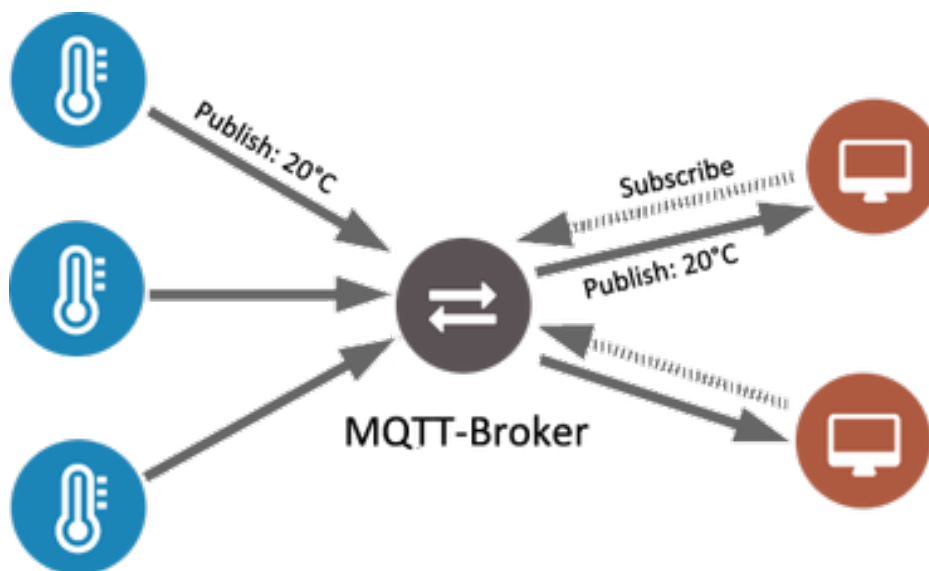


Рисунок 3.1 – Приклад роботи MQTT брокера

Для встановлення MQTT брокера Mosquitto потрібно увімкнути Raspberry Pi та відкрити термінал. Тепер у терміналі прописуємо команду для оновлення системи Raspberry:

```
sudo apt-get update
```

Після чого встановлюємо сам брокер командою:

```
sudo apt-get install mosquitto
```

Також встановимо MQTT клієнт:

```
sudo apt-get install mosquitto mosquitto-clients
```

Після встановлення mosquitto сервер буде автоматично запускатись при ввімкненні системи.

Для керування системою за допомогою мови програмування Python у середовищі PyCharm було розроблено програму з графічним користувацьким інтерфейсом. Графічний інтерфейс програми реалізовано за допомогою графічної бібліотеки Python Tkinter[1]. Інтерфейс програми зображено на рис. 3.2. Дана програма буде працювати на Raspberry Pi 3.

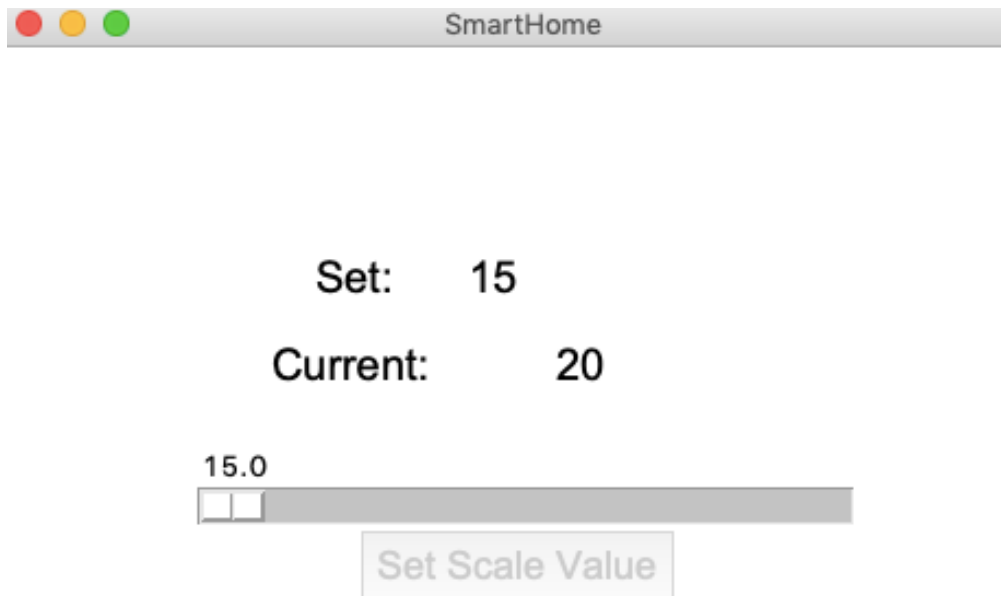


Рисунок 3.2 – Інтерфейс програми

Для реалізації програми керування системою окрім бібліотеки Tkinter також використовується MQTT бібліотека[7] для спілкування пристроїв в системі. Код реалізації інтерфейсу зображено на рис. 3.3.

```

window = Tk()
window.title("SmartHome")
window.geometry('|'480x320')
switchModel = Button(window, text="Set Scale Value",
background="#555", foreground="#ccc",
padx="5", pady="5", font="Arial 18",command =
sel)
switchModel.place(x = 165, y = 220)
lblTemp = Label(window, font="Arial 20", text = tempDef)
lblTemp.place(x = 250, y = 130)
lblNameCurTemp = Label(window, font="Arial 20", text =
"Current: ")
lblNameCurTemp.place(x = 120, y = 130)
lblNameSetTemp = Label(window, font="Arial 20", text = "Set:
")

```

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

```

lblNameSetTemp.place(x = 140, y = 90)
lblNameSetTempLbl = Label(window, font="Arial 20", text =
scaleData)
lblNameSetTempLbl.place(x = 210, y = 90)
var = DoubleVar()
scale = Scale(window, variable = var, length = 300, orient =
HORIZONTAL, resolution=0.1, from_ = 15, to = 30)
scale.place(x = 90, y = 180)
window.mainloop()

```

Рисунок 3.3 – Лістинг реалізації інтерфейсу

Функція для отримання даних про температуру в приміщенні реалізовано асинхронно від всієї програми за допомогою Python-бібліотеки `asyncio`. Таким чином функція працює в фоновому режимі та отримує значення температури коли модуль температури відправить значення температури. Код асинхронної функції отримання значень температури зображено рис. 3.4.

```

async def getData():
    client = mqtt.Client("digi_mqtt_test")
    client.username_pw_set(username="username",
password="password")
    client.on_connect = on_connect
    client.on_message = on_message
    client.connect('192.168.1.103', 1883)
    client.loop_start()
loop = asyncio.get_event_loop()
loop.run_until_complete(getData())
loop.close()

```

Рисунок 3.4 – Лістинг асинхронної функції

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Відправлення даних про температуру, статус радіатора та значення встановленої температури на телефон. Код реалізації функції відправлення даних по протоколу MQTT зображено на рис. 3.5.

```
def sendData(topic, data):
    clientSend = mqtt.Client("set/Temperature")
    clientSend.username_pw_set(username="username",
password="password")
    clientSend.connect('192.168.1.103', 1883)
    time.sleep(0.5)

    if topic == "set/Temp":
        clientSend.publish(topic="set/Temp", payload=data)
        clientSend.disconnect()
    elif topic == "set/Status":
        clientSend.publish(topic="set/Status", payload=data)
        clientSend.disconnect()
    elif topic == "get/ScaleToPhone":
        clientSend.publish(topic="get/ScaleToPhone", payload=data)
        clientSend.disconnect()
```

Рисунок 3.5 – Лістинг функції відправлення даних

Інтерфейс розробленого додатку реалізовано за допомогою Interface Builder. Вигляд розробленого інтерфейсу зображено на рис. 3.6.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

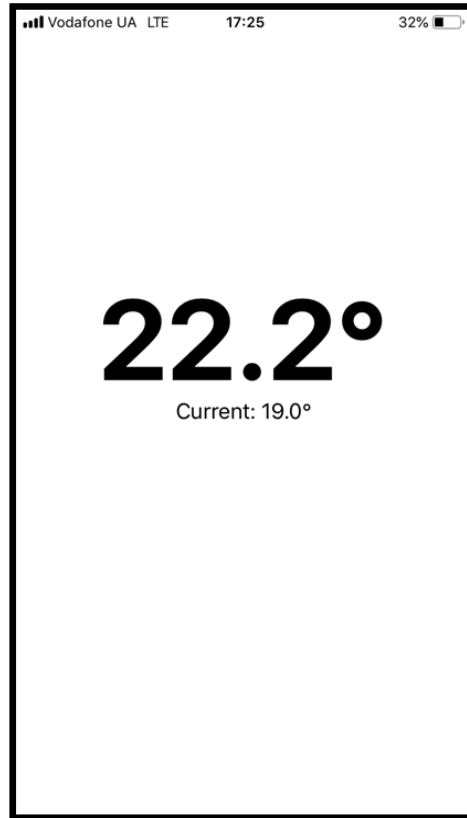


Рисунок 3.6 – Інтерфейс мобільного додатку

Для роботи з MQTT брокером у Swift було встановлено бібліотеку CocoaMQTT[4]. Ця бібліотека встановлюється за допомогою CocoaPods[9] – це кращий засіб з управління залежностями бібліотек в Objective-C проектах. Замість скачування коду зі сховищ бібліотеки і копіювання в папку вашого проекту ми можемо надати можливість CocoaPods зробити це самостійно.

Для початку роботи потрібно встановити CocoaPods. Зробити це можна за допомогою команд:

```
sudo gem install cocoapods &&
pod setup
```

Тепер для роботи з Pod потрібно у терміналі перейти у папку проекту та прописати команду

```
pod init
```

За допомогою цієї команди в папці проекту створюється файл pod. Потрібно зайти в нього та прописати всі бібліотеки які потрібно встановити у цей проект. Після чого у терміналі прописуємо команду

```
pod install
```

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Після чого в папці проекту створюється файл з розширенням `xcworkspace`. Цей файл слід відкривати у середовищі розробки програмного забезпечення Xcode. Цей файл буде вже містити встановлені бібліотеки які готові до використання.

Для початку роботи з MQTT потрібно задати настройки підключення у коді. Функція налаштування з'єднання з брокером MQTT знаходиться на рис. 3.7.

```
funk setumot() {
    let clientID = "CocoaMQTT-" +
String(ProcessInfo().processIdentifier)
    mqtt = CocoaMQTT(clientID: clientID, host:
"192.168.1.103", port: 1883)
    mqtt.username = "username"
    mqtt.password = "password"
    mqtt.willMessage = CocoaMQTTWill(topic: "/will",
message: "dieout")
    mqtt.keepAlive = 60
    mqtt.delegate = self
    mqtt.connect()
}
```

Рисунок 3.7 – Лістинг функції налаштування з'єднання з MQTT брокером

Для отримання даних про температуру у приміщенні та для зчитування даних про встановлену температуру на панелі керування системою було реалізовано метод з бібліотеки `CocoaMQTT` яка зображена на рис. 3.8.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

```

func mqtt(_ mqtt: CocoaMQTT, didReceiveMessage message:
CocoaMQTTMessage, id: UInt16) {
    print("Message recived: \(message.string ?? "nil") |
topic: \(message.topic)")
    if message.topic == "get/Temp" {
        if let data =
Float(message.string!.suffix(5).prefix(4)) {
            temperature.currentTemperature = data
        }
    } else if message.topic == "get/ScaleToPhone" {
        if let data = Float(message.string!) {
            temperature.settedTemperature = data
            settedTemperatureLabel.text = String(format:
"% .1f°", data)
        }
    }
}
}

```

Рисунок 3.8 – Лістинг методу CocoaMQTT для зчитування даних

Для відправлення значення температури якої потрібно досягти системі було реалізовану функцію, код якої відображено на рис. 3.9.

```

func settedTemperatureWasUpdated(value: Float) {
    print("Setting needed temperature: \(String(format:
"% .1f°", value)) °")
    mqtt!.publish("get/Scale", withString: String(format:
"% .1f", value)) // Send needed temperature
}

```

Рисунок 3.9 – Лістинг функції для відправлення встановленої температури

Для комфортного користування додатком було реалізовано можливість встановлення температури методом переміщення курсора по екрану. Під час ковзання пальця по екрану виконується метод, який регулює значення

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

температури, яку необхідно досягнути. Код реалізації цього методу зображено на рис. 3.10.

```
        let gesture = UIPanGestureRecognizer(target: self, action:
#selector(ViewController.temperatureSlider(_:)))
        view.addGestureRecognizer(gesture)

        @objc func temperatureSlider(_ sender: UIPanGestureRecognizer)
        {
            let velocity = sender.velocity(in: view)

            if velocity.y > 0 { temperature.value -=
Float(velocity.y * 0.0002) } // UP
            else { temperature.value -= Float(velocity.y * 0.0002)
} // BOTTOM

            if sender.state == .changed {
                self.temperature.value =
self.temperature.settedTemperature
                settedTemperatureLabel.text = String(format:
"%1f°", temperature.settedTemperature)
            } else if sender.state == .ended {

temperature.mqttDelegate.settedTemperatureWasUpdated(value:
temperature.settedTemperature)
            }
        }
    }
}
```

Рисунок 3.10 – Лістинг методу регулювання температури якої необхідно досягнути

Вигляд комплектуючих в середині корпуса готового модуля радіатора зображено на рис. 3.11

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

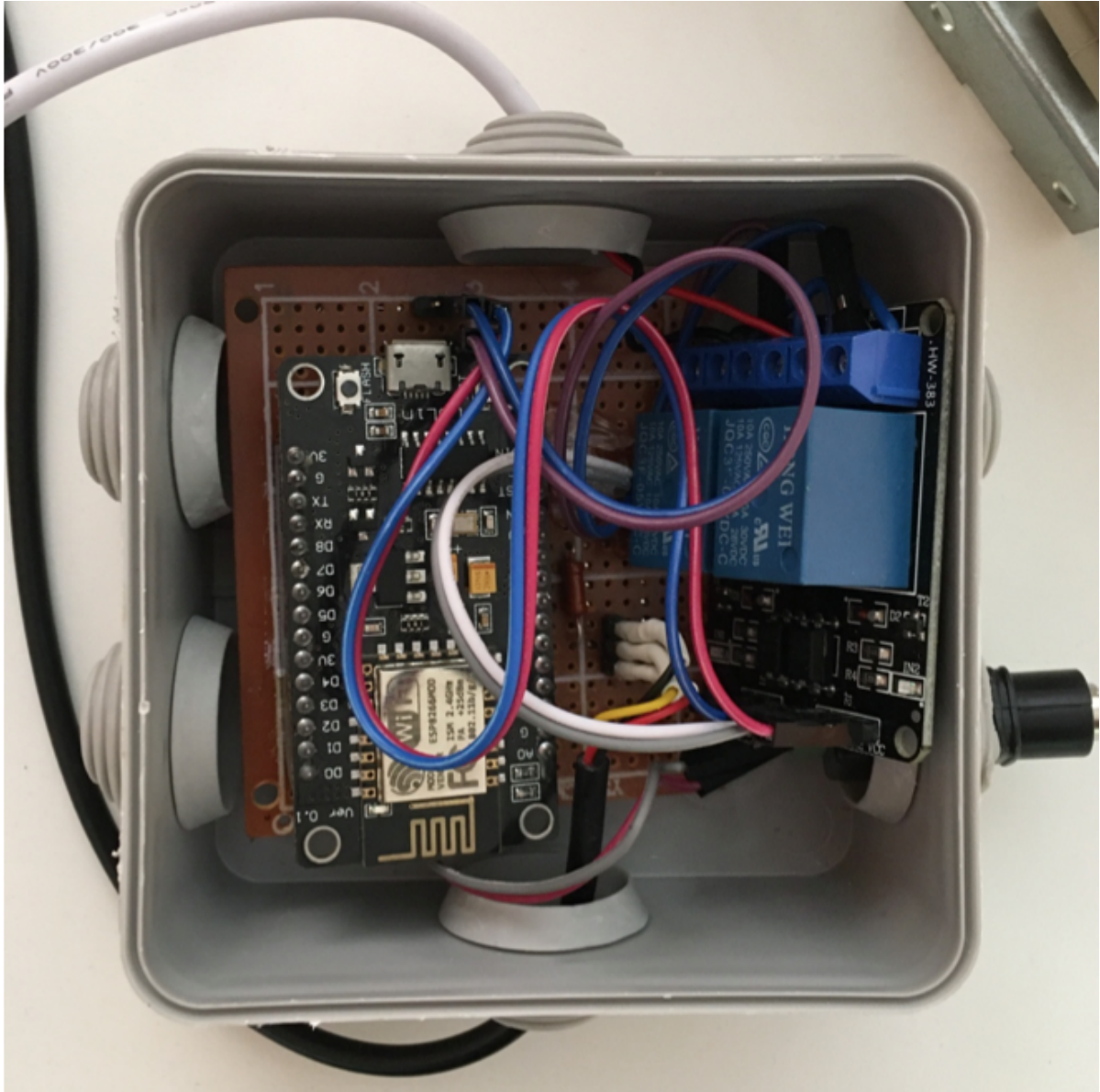


Рисунок 3.11 – Комплектуючі готового модуля радіатора

Вигляд всієї готової до використання КС КРО. зображено на рис. 3.12

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.175.00.00 ПЗ

Арк.

33

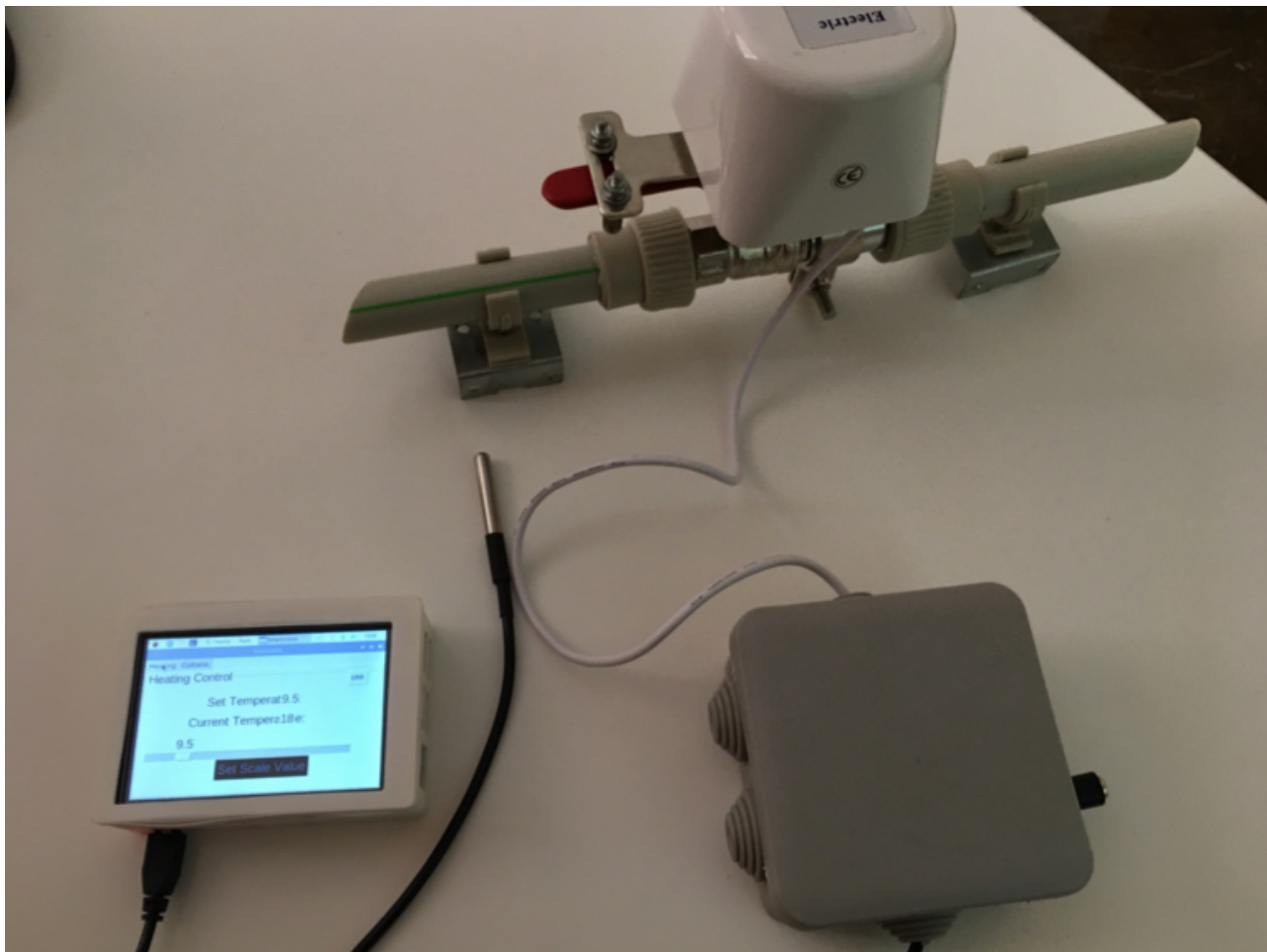


Рисунок 3.12 – Загальний вигляд КС КРО

3.2 Тестування

Для початку роботи системи потрібно запустити Raspberry Pi (панель керування). На ній автоматично запуститься графічний інтерфейс (рис. 3.13).



Рисунок 3.13 – Панель керування

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Після того як панель керування запущено запускаємо модуль вимірювання температури (рис. 3.14).

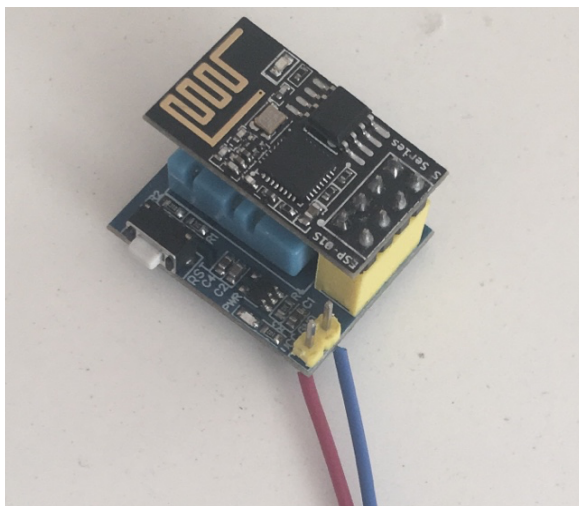


Рисунок 3.14 – Модуль вимірювання температури

Після включення він буде працювати в режимі WI-FI точки доступу з назвою «TMP-ESP» та паролем «12345678» (рис. 3.15). Потрібно підключитись до цієї точки та у веб-браузері перейти на його адресу, а саме 192.168.4.1. Після чого буде запущена веб-сторінка на якій потрібно ввести назву мережі та пароль від мережі до якої вже підключено панель керування (рис. 3.16).

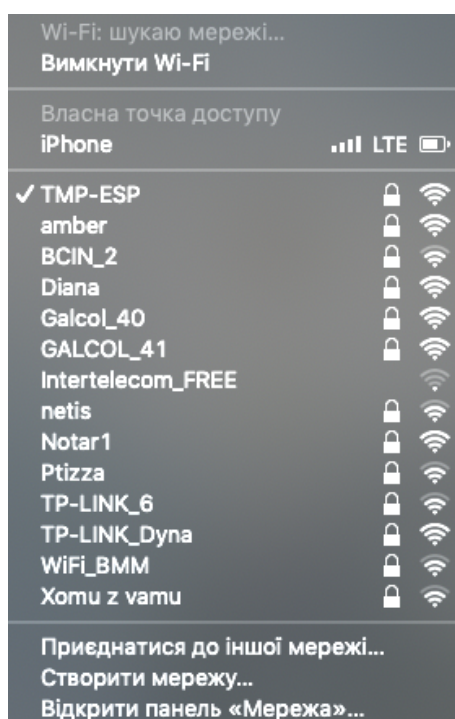


Рисунок 3.15 – Налаштування точки доступу модуля температури

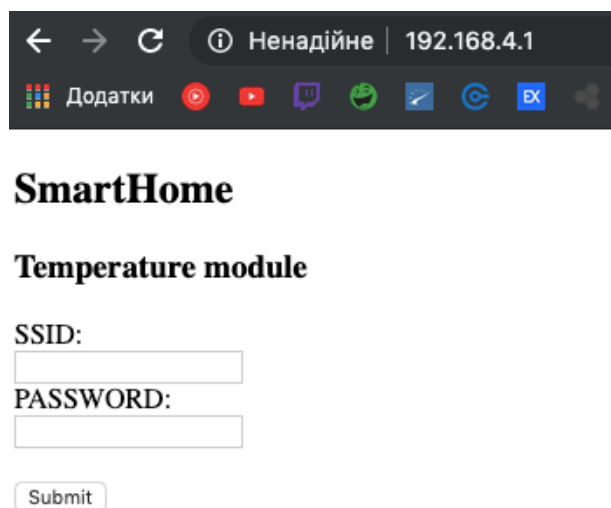


Рисунок 3.16– Веб-сервер модуля вимірювання температури

Після цього температурний модуль підключиться до домашньої мережі та буде надсилати покази на панелі керування. Яка буде виводити ці дані на екрані (рис. 3.17).

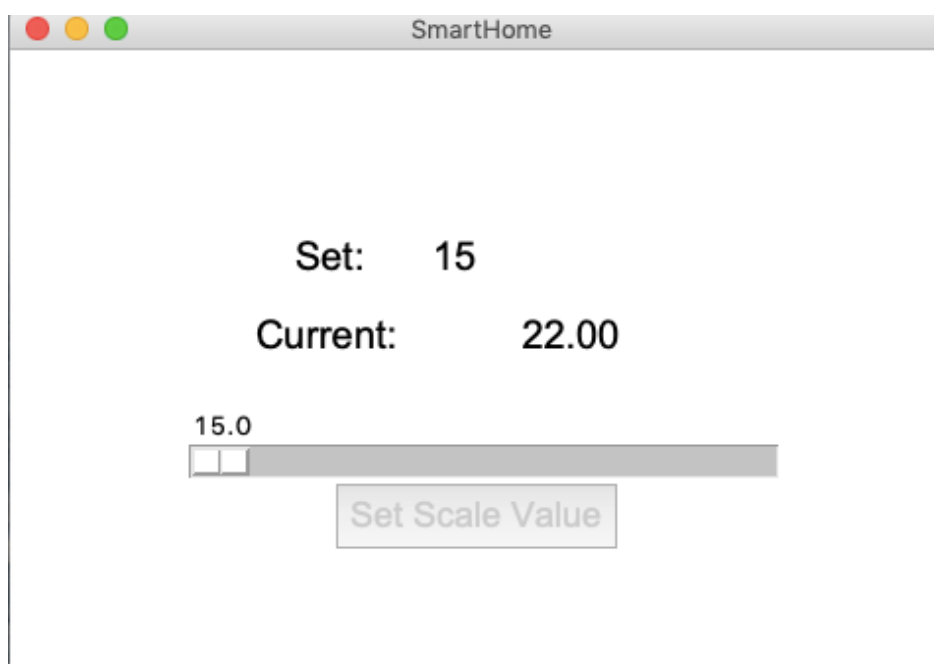


Рисунок 3.17 – Виведення поточної температури після підключення модуля вимірювання температури

Після запуску модуля вимірювання температури запускаємо модуль керування радіатором. Вигляд модуля керування радіатором зображено на рис. 3.18.



Рисунок 3.18 – Модуль керування радіатором

Модуль керування радіатором так як і модуль температури також запускається в режимі точки доступу. Для налаштування модуля керування радіатором потрібно підключитись до Wi-Fi мережі «Battery-ESP» та ввести пароль «12345678». Далі у веб-браузері перейти по адресі 192.168.1.103 та ввести дані від мережі, до якої підключені інші компоненти системи (рис. 3.19).

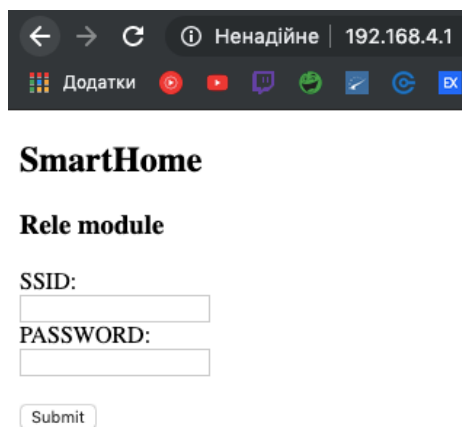


Рисунок 3.19 – Веб-сервер модуля керування радіатором

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Тепер система готова до використання.

Для регулювання температури у приміщені окрім панелі керування також можна використати створений мобільний додаток (рис. 3.3). Температура, яка буде встановлена у мобільному додатку, також синхронізується з панеллю керування на Raspberry Pi: якщо температуру виставлено на панелі керування Raspberry Pi, тоді і у мобільному додатку встановлюється також ця температура. Отже, для початку встановимо переглянемо поточну температуру у приміщені. Ми бачимо, що температура у приміщені 24 градуси, для початку встановимо температуру з панелі керування на охолодження, а саме на 21 градус (рис. 3.20).

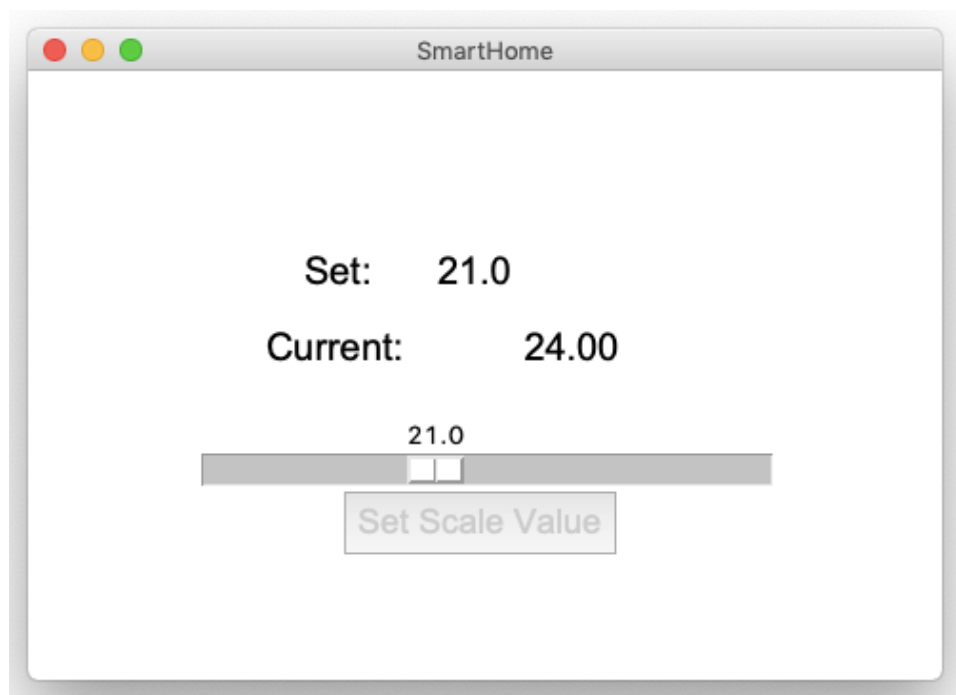


Рисунок 3.20– Встановлена температура з панелі керування

Також всі дані синхронізуються з мобільним додатком для зручності керування системою. На рис. 3.21 зображено, що температура, яка встановлена на панелі керування синхронізується і у мобільному додатку.

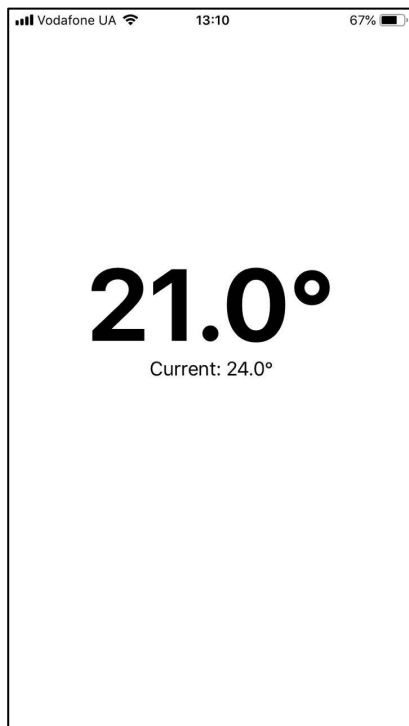


Рисунок 3.21 – Синхронізація даних у мобільному додатку з панеллю керування

Після того, як температуру встановили на зниження потік води у радіаторі перекривається. Очікуємо поки температура у приміщені не буде нижча. Ми бачимо, що температура у приміщені через деякий час починає зменшуватись, тому встановимо даліше для прикладу встановимо у мобільному додатку температуру у приміщені вищу від поточної (рис. 3.22).

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

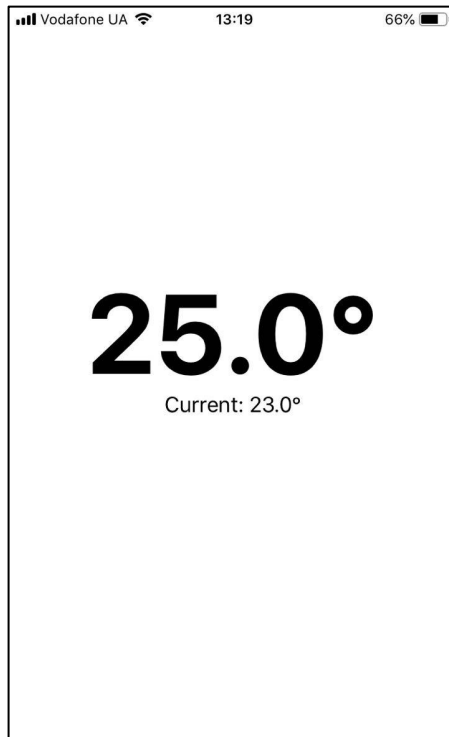


Рисунок 3.22 – Встановлена температура через мобільний додаток

Також температура, яка була встановлена через мобільний додаток синхронізується з панеллю керування.

Після того, як температуру встановили вищу від поточної температури у приміщені, вентиль на водяному радіаторі відкривається для того, щоб він радіатор нагрівався тим самим піднімав температуру у приміщені.

Протестувавши КС КРО стає зрозуміло, що система готова до використання на практиці.

					КС КРБ 123.175.00.00 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

/

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська допомога при опіках

При наданні домедичної допомоги розрізняють опіки чотирьох ступенів:

I ступінь (еритема) - почервоніння шкіри, набряклість і біль;

II ступінь (утворення пухирів) - сильний біль із інтенсивним почервонінням, відшаруванням епідермісу з утворенням міхурів, наповнених прозорою або каламутною рідиною;

III ступінь: некроз всієї товщі шкіри з утворенням щільного струпу, під яким перебувають ушкоджені тканини;

IV ступінь (обвуглення): виникає при впливі на тканини дуже високих температур (полум'я, розплавлений метал тощо); частіше при пожежах та аваріях на автотранспорті (ДТП), в літаках, нещасні випадки на шахтах; результат таких опіків - ушкодження м'язів, сухожилів, кісток[1].

Послідовність дій при наданні домедичної допомоги постраждалим з опіками:

- 1) переконатися у відсутності небезпеки;
- 2) провести огляд постраждалого, визначити наявність свідомості, дихання;
- 3) викликати бригаду екстреної (швидкої) медичної допомоги;

					КС КРБ 123.175.00.00 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.					БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	Літ.	Арк.	Акрушів
Перевір.							42	5
Реценз.						ТНТУ, каф. КС, гр. СІс-44		
Н. контр.								
Затверд.								

4) якщо у постраждалого відсутнє дихання, розпочати проведення серцево-легеневої реанімації;

5) якщо у постраждалого опіки першого і/або другого ступеня:

— охолодити місце опіку прохолодною водою;

— після охолодження накрити пошкоджену ділянку чистою вологою серветкою; не слід спеціально проколювати пухирі; якщо пухирі розірвались, накласти чисту, стерильну пов'язку;

4.2 Розробка заходів щодо зниження рівня шуму на дільниці цеху

Впровадження у виробництво нових технологічних процесів, підвищення потужностей і швидкохідності технологічного устаткування, механізація виробничих процесів призвели до того, робітники протягом зміни більшою чи меншою мірою зазнають негативного впливу шуму.

Шум — одна з найпоширеніших виробничих шкідливостей, яку дуже складно усунути. При тривалому впливі шуму не тільки знижується гострота слуху, але й погіршується робота центральної нервової і серцево-судинної систем, шлунково-кишкового тракту, органів дихання, виникають запаморочення і функціональні зміни нервової системи, в тому, ослаблення пам'яті та уваги. Шум може стати причиною виробничого травматизму та зниження продуктивності праці: так, підвищення рівня шуму на 10 дБ зменшує продуктивність праці на 10%. Увесь комплекс змін, який відбувається в організмі людини при тривалому впливі шуму, розглядається як так звана "шумова хвороба"[16].

Проектування захисту від шуму в цехах і на території промислових підприємств полягає у:

Визначенні за результатами акустичного розрахунку (або за результатами інструментальних вимірювань на діючих підприємствах) необхідного зниження рівнів шуму на робочих місцях і зонах постійного перебування виробничого

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

персоналу в приміщеннях з джерелами шуму, на робочих місцях і зонах постійного перебування персоналу на території підприємства, в приміщеннях без власних джерел, але які потребують захисту від шуму обладнання, встановленого в іншому приміщенні або в іншій будівлі, та розробленні на основі визначеного необхідного зниження рівнів шуму комплексу заходів щодо зниження рівнів шуму до нормативних величини як в межах підприємства, так і на прилеглих до нього територіях з нормованими рівнями шуму.

Заходи та засоби захисту від шуму поділяються на колективні та індивідуальні, причому останні застосовуються лише тоді, коли заходами та засобами колективного захисту не вдається знизити рівні шуму на робочих місцях до допустимих значень. Призначення засобів індивідуального захисту від шуму - перекрити найбільш чутливі канали проникнення звуку в організм - вуха. Тим самим різко послаблюються рівні звуків, що діють на барабанну перетинку, а відтак - і коливання чутливих елементів внутрішнього вуха. Такі засоби дозволяють одночасно попередити розлад і всієї нервової системи від дії інтенсивного подразника, яким є шум.

До індивідуального захисту від шуму належать навушники, протишумові вкладки, шумозаглушувальні шоломи. Такий вибір обумовлюється видом та характеристикою шуму на робочому місці, зручністю використання засобу при виконанні даної робочої операції та конкретними кліматичними умовами.

Засоби колективного захисту від шуму подібно до віброзахисту поділяються за такими напрямками:

- зменшення шуму в самому джерелі;
- зменшення шуму на шляху його поширення;
- організаційно-технічні заходи;
- лікувально-профілактичні заходи.

Зменшення шуму в самому джерелі - найбільш радикальний засіб боротьби з шумом, що створюється устаткуванням. Ефективність заходів щодо зниження шуму устаткування, що вже працює, досить невисока, тому необхідно прагнути

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

до максимального зниження шуму в джерелі ще на стадії проектування устаткування.

Для зменшення шкідливого впливу виробничого шуму на працівників шумних виробництв, послаблення передавання його в сусідні приміщення застосовують звуко- і віброізоляцію, звуко- і вібропоглинання та глушники шуму.

Звукоізоляція є ефективним засобом зменшення рівня шуму у напрямку його поширення, що реалізується шляхом встановлення звукоізоляційних перешкод (перегородок, кабін, кожухів, екранів). Принцип звукоізоляції базується на тому, що більша частина звукової енергії, яка потрапляє на перешкоду, відбивається і лише незначна її частина проходить крізь неї.

Для звукоізоляції окремих шумних діляниць у приміщенні чи устаткування застосовують легкі багатошарові звукоізоляційні перегородки з повітряними прошарками. Для звукоізоляції найбільш шумних вузлів та агрегатів (ланцюгові передачі, двигуни, компресори, вентилятори) використовуються звукоізоляційні кожухи, які є засобами, що встановлюються в безпосередній близькості від джерела шуму. В тих випадках, коли неможливо ізолювати шумне устаткування чи його вузли, захист працівника від дії шуму здійснюють шляхом облаштування звукоізолюваної kabіни з пультом керування та оглядовими вікнами.

Метод акустичного екранування застосовується в тих випадках, коли інші методи малоефективні або недоцільні з техніко-економічної точки зору. Акустичний екран встановлюється між джерелом шуму та робочим місцем і являє собою певну перешкоду на шляху поширення прямого шуму, за якою виникає так звана звукова тінь

4.3 Трудова діяльність користувачів комп'ютерних технологій.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

Охорона праці (ОП) - система законодавчих актів, соціально-економічних, організаційних, технічних, гігієнічних, лікувально профілактичних заходів, направлених на: забезпечення безпеки людини, збереження здоров'я та працездатності людини в процесі праці; розробку методів і засобів захисту шляхом зниження впливу шкідливих і небезпечних чинників до допустимих значень. Завдання ОП - звести до мінімуму вірогідність травмування або захворювання працюючого з одночасним забезпеченням комфорту для максимальної продуктивності праці.[17]

Найпоширенішими у процесі праці є поза сидяча і стояча. При організації робочого місця потрібно враховувати, що при виконанні роботи з фізичними навантаженнями бажаною є "стояча" поза, а при малих зусиллях "сидяча".

Робоча поза "стоячи" призводить до більшої втоми, ніж "сидячи". На одну і ту ж роботу людина витрачає на 10 % більше енергії, що призводить до підвищення артеріального і венозного тиску крові, розширення вен на ногах, тромбофлебиту, пошкодження стоп, викривлення хребта.

При сидячій роботі нижня частина тіла розслаблена, а основне статичне навантаження припадає на м'язи шиї, спини, таза, стегон. Неправильна сидяча поза може викликати застій крові в ногах, а при великому обсязі робіт руками до запалення суглобів.

Робота на ПК - модель розумової праці, що тривалий час виконується в одноманітній позі, в умовах обмеження загальної м'язової активності і за рухливості рук, тривалого високого напруження зорових функцій, нервово-емоційного напруження, в умовах впливу різноманітних фізичних чинників, які по різному впливають на людину.

Отже, небезпечні фактори погіршують умови праці надаючи шкідливу дію на організм людини. Наприклад, ті хто працюють в умовах тривалої шумової дії відчувають дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену стомлюваність, пониження апетиту, болі у вухах і т.д.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

ВИСНОВКИ

Здійснивши аналіз предметної області було визначено мету розробки системи, сформульовано та обґрунтовано задачу, функції системи та її будова, окреслено вхідні та вихідні дані та визначено вимоги до системи.

На першому кроці було запроектовано комп'ютеризовану систему керування радіаторним опаленням, яка здатна в автоматичному режимі підтримувати задану температуру у приміщенні, керуючись з сенсорної панелі центрального модуля системи або мобільного додатку.

На наступному кроці було визначено засоби вирішення поставленої задачі. Зокрема, було обрано платформу ESP для вирішення цієї задачі та ряд спеціальних модулів та сенсорів, а саме температури які були використані для реалізації модулів температури та радіаторів.

В результаті роботи було реалізовано та протестовано на практиці вище зазначену систему.

Так, як система є досить гнучкою для нарощування нових функцій. Це дає дуже великий потенціал для покращення системи та масштабування її на більші приміщення. Також, можна розширювати функціонал за рахунок додавання нових модулів та сенсорів. Або ж синхронізувати дану систему з вже готовими на ринку системами тип «Розумний» будинок.

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

БІБЛІОГРАФІЯ

1. Alan D. Moore. Python GUI programming with the Tkinter : Видавництво: O'Reilly Media, 2015 (дата звернення: 01.03.2019).
2. Mosquitto MQTT Broker, *Steve's Internet Guide* : веб-сайт. URL: <http://www.steves-internet-guide.com/mosquitto-broker/> (дата звернення: 01.02.2019).
3. Rayan Knightly, MQTT In iOS, *Medium* : веб-сайт URL: <https://medium.com/thefloatingpoint/mqtt-in-ios-d8574b55e006> (дата звернення: 10.04.2019).
4. Електромагнітне реле : веб-сайт. URL: <https://www.arhivinfo.ru/1-32775.html> (дата звернення 14.05.2021)
5. Swift. *Developer* : веб-сайт. URL: <https://developer.apple.com/swift/> (дата звернення: 30.03.2019).
6. 3D - моделювання в AutoCAD : веб-сайт. URL: <https://uk.soringpcrepair.com/3d-modeling-in-autocad/> (дата звернення: 26.03.2019).
7. Використання MQTT з Raspberry Pi і ESP8266, *Instructables* : веб-сайт. URL: <https://www.instructables.com/id/How-to-Use-MQTT-With-the-Raspberry-Pi-and-ESP8266/> (дата звернення: 05.02.2019).
8. Прошивка ESP8266 через Arduino IDE, *Arduinomaster* : веб-сайт. URL: <https://arduinomaster.ru/platy-arduino/proshivka-esp8266-cherez-arduino-ide/> (дата звернення: 25.03.2019).
9. Электрический привод крана, *MySku* : веб-сайт. URL: <https://mysku.ru/blog/aliexpress/66896.html> (дата звернення 28.03.2021).
10. AKhatmullin, Основы использования Socoarods в разработке приложений под iOS, *Habr* : веб-сайт. URL: <https://habr.com/ru/post/261711/> (дата звернення: 02.04.2019).

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

11. Управління опаленням майбутнього, Твоє Тепло : веб-сайт. URL: <https://tvoeteplo.com.ua/upravlinnya-opalennyam-majbutnogo/> (дата звернення 10.05.2021).
12. Термогігрометри від SIMVOLT, Simvolt : веб-сайт. URL: <https://simvolt.ua/termogigrometri-vid-simvolt-kontroluemo-temperaturu-i-vologist/> (дата звернення 12.05.2021).
13. Проектування комп'ютерних систем, Кластер : веб-сайт. URL: <https://klaster.ua/ua/services/kompjuternye-sistemy/proektirovanie-i-soglasovanie/> (дата звернення 14.05.2021).
14. Програмування Arduino, Arduino.UA : веб-сайт. URL: <https://doc.arduino.ua/ru/prog/> (дата звернення 15.05.2021).
15. Он-лайн сервіс EasyEDA, Chema.net : веб-сайт. URL: <https://schem.net/software/easyeda.php> (дата звернення: 20.05.2021)
16. ДСТУ Н.Б.В 1.1-35:2013 «Настанова з розрахунку рівнів шуму в приміщеннях і на територіях»
17. ДБН В.1.2-8:2008 „Основні вимоги до будівель і споруд. Безпека життя

					КС КРБ 123.175.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

ДОДАТКИ

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

“Затверджую”

Завідувач кафедри КС

_____ Осухівська Г.М.

“ ___ ” _____ 2021 р

КОМП'ЮТЕРИЗОВАНА СИСТЕМА КЕРУВАННЯ РАДІАТОРНИМ
ОПАЛЕННЯМ

ТЕХНІЧНЕ ЗАВДАННЯ

на 4 листках

Вид робіт:

Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»

«ВИКОНАВЕЦЬ»

Керівник кваліфікаційної роботи

Студент групи СІс-44

_____ к.т.н., доц. Осухівська Г.М.

_____ Мікула О.О.

« ___ » _____ 2021 р.

« ___ » _____ 2021 р.

Тернопіль 2021

1. Назва та підстава для виконання роботи.

1.1. Комп'ютерна система керування радіаторним опаленням.

1.2. Підставою для виконання кваліфікаційної роботи бакалавра (КРБ) є

Наказ по Університету (№ 4/7-59 від 01.02.2021 р.).

2. Виконавець.

2.1. Студента групи СІс-44 кафедри КС

Тернопільського національного технічного університету ім. І. Пулюя

Мікула Олег Орестович.

3. Мета роботи.

3.1. Метою роботи є розробити структуру та апаратне забезпечення комп'ютерної системи керування радіаторним опаленням.

4. Склад виробу.

4.1. До складу системи повинні входити:

- 1) мікроконтролер;
- 2) датчик температури;
- 4) реле або виконавчі пристрої;
- 5) клавіші керування;

б) індикатори для відображення температури;

8) мотор модуля радіатора

5. Технічні вимоги.

5.1. Вимоги по призначенню.

5.1.1. Комп'ютерна система повинна мати наступні параметри:

1) Діапазон вимірюваної температури, °С -20...+80

3) Точність вимірювання температури, °С ±0,5

5.1.2. Швидкість обміну даними по шині I²C, не менше кбіт/с 1000

5.1.3. Пристрій повинен живитись напругою постійного струму, В +12±2

5.2. Вимоги до умов експлуатації:

5.2.1. По умовам експлуатації виріб повинен відповідати вимогам ГОСТ 15150 для УХЛ4.1

5.2.2. Температура експлуатації від -30 до +80°C

5.2.3. Відносна вологість до 100% при t=25°C

5.3. Конструктивні вимоги.

5.3.1. Конструювання корпусу приладу в КРБ не передбачено.

5.3.2. Для побудови системи мають бути використані сучасні компоненти з можливістю поверхневого монтажу друкованого вузла.

5.3.3. При побудові системи необхідно передбачити розміщення роз'ємів живлення і обміну даними.

5.3.4. Габаритні розміри при макетуванні, мм, не більше:

довжина	200
ширина	300
висота	120
5.3.5. Маса макету, кг, не більше	2,5

5.3.6. Конструкція макету повинна забезпечувати доступ до всіх комплектуючих виробів при тестуванні.

5.4. Вимоги до надійності.

5.4.1. Система повинна відповідати вимогам ДСТУ 2862-94.

5.4.2. Наробка на відмову, не менше 5300 год.

5.5. Вимоги метрології.

5.5.1. Вимірювання параметрів системи при моделюванні повинно виконуватись на універсальних вимірювальних приладах.

6. Економічні показники.

6.1. Собівартість системи повинна бути не більше 8000 грн.

7. Вимоги до документації.

7.1. Конструкторська документація повинна відповідати вимогам ЄСКД, ДСТУ та ГОСТ.

7.2. До складу документації повинно входити:

- 1) ПЗ
- 2) Структурна схема Е1

- 3) Електрична схема ЕЗ
- 4) Схема з'єднань Е4
- 5) Блок схема алгоритму роботи

8. Стадії та етапи розробки КРБ

8.1 Стадії та етапи виконання КРБ наведенні в таблиці 1.

Таблиця 1

№	Назва етапу	Строк виконання	
		початок	кінець
1	Технічне завдання	—	до 31.03.21
2	Розділ 1	01.04. 21	09.04. 21
3	Розділ 2	09.04. 21	01.05. 21
4	Розділ 3	02.05. 21	15.05. 21
5	Розділ охорона праці	16.05. 21	24.05. 21
6	Нормоконтроль	25.05. 21	14.06. 21
7	Попередній захист	14.06. 21	22.06.21
8	Захист	з 26.06. 21	—

9. В дане ТЗ можуть вноситись зміни по узгодженню сторін

Додаток Б

Лістинги реалізації модулів комп'ютеризованої системи керування радіаторним опаленням

Лістинг Б1 - Лістинг реалізації модуля передачі температури

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <PubSubClient.h>
#include <dht.h>
#define DHTTYPE DHT11

String loginForWiFi = "";
String passwordForWiFi = "";
bool flag = false;
bool flag2 = false;
const char* mqtt_server = "192.168.1.103";
const char* keyDevice = "DHTMODULO1";
const char* mqttUser = "username";
const char* mqttPassword = "password";
unsigned long timing;
int DHTPIN = 2;
float temp = 0;
char msgBuffer[100];

// WiFi network
const char* ssid = "TMP-ESP";
const char* password = "12345678";
ESP8266WebServer server ( 80 );
WiFiClient espClient;
PubSubClient client(espClient);
dht DHT;
```

```

char htmlResponse[3000];
const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<body>
<h2>SmartHome</h2>
<h3>Temperature module</h3>
<form action="/action_page">
  SSID:<br>
  <input type="text" name="loginForWiFi" value="">
  <br>
  PASSWORD:<br>
  <input type="text" name="passwordForWiFi" value="">
  <br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
)=====";

void handleRoot() {
  String s = MAIN_page;
  server.send ( 200, "text/html", s);
}

void handleSave() {
  if (server.arg("loginForWiFi") != "") {
    Serial.println("SSID: " + server.arg("loginForWiFi"));
    loginForWiFi = server.arg("loginForWiFi");
    flag = true;
  }
  if (server.arg("passwordForWiFi") != "") {
    passwordForWiFi = server.arg("passwordForWiFi");
    Serial.println("PASSWORD: " + passwordForWiFi);
  }
  String s = "<a href='/'> Go Back </a>";
  server.send(200, "text/html", s);
}

```

```

}

void reconnect() {
  while (!client.connected()) {
    if (client.connect("dht11modulo", mqttUser, mqttPassword))
    {
      client.publish("test", keyDevice);
      client.subscribe(keyDevice);
    } else {
      delay(5000);
    }
  }
}

void sendTemp() {
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  if (!client.connected()) {
    reconnect();
    client.publish("casa", keyDevice);
  }
  while (true) {
    int readData = DHT.read11(DHTPIN);
    temp = DHT.temperature;
    delay(1000);
    Serial.println("temp:");
    temp = temp - 5;
    Serial.println(temp);
    client.publish("get/Temp", dtostrf(temp, 6, 2, msgBuffer));
    delay(6000);
  }
}

void callback(char* topic, byte* payload, unsigned int length)
{
}

```

```

void setup() {
  Serial.begin(9600);
  Serial.println("");
  WiFi.mode(WIFI_AP_STA);
  WiFi.softAP(ssid, password); //Start HOTspot removing
password will disable security
  IPAddress myIP = WiFi.softAPIP(); //Get IP address
  Serial.print("HotSpt IP:");
  Serial.println(myIP);
  server.on("/", handleRoot); //Which routine to handle at
root location
  server.on ("/action_page", handleSave);
  server.begin(); //Start server
  Serial.println("HTTP server started");
}

```

```

void loop() {
  if (flag == false) {
    server.handleClient();
  }
  if (flag == true) {
    int numberOfNetworks = WiFi.scanNetworks();
    for (int i = 0; i < numberOfNetworks; i++) {
      if (loginForWiFi == WiFi.SSID(i)) {
        Serial.println(WiFi.SSID(i));
        WiFi.softAPdisconnect (true);
        WiFi.begin(loginForWiFi, passwordForWiFi);
        Serial.println(WiFi.waitForConnectResult());
        if (WiFi.waitForConnectResult() == WL_CONNECTED) {
          Serial.println("Connect good");
          sendTemp();
        } else {
          Serial.println("Password faild");
          WiFi.softAP(ssid, password);
          flag = false;
        }
      }
    }
  }
}

```



```

const char* ssid      = "Battery-ESP";
const char* password = "12345678";
ESP8266WebServer server ( 80 );
WiFiClient espClient;
PubSubClient client(espClient);

char htmlResponse[3000];
const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<body>

<h2>SmartHome</h2>
<h3>Rele module</h3>

<form action="/action_page">
  SSID:<br>
  <input type="text" name="loginForWiFi" value="">
  <br>
  PASSWORD:<br>
  <input type="text" name="passwordForWiFi" value="">
  <br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
)=====";

void handleRoot() {
  String s = MAIN_page;
  server.send ( 200, "text/html", s);
}

void handleSave() {
  if (server.arg("loginForWiFi") != "") {
    Serial.println("SSID: " + server.arg("loginForWiFi"));
  }
}

```



```

    loginForWiFi = server.arg("loginForWiFi");
    flag = true;
}
if (server.arg("passwordForWiFi") != "") {
    passwordForWiFi = server.arg("passwordForWiFi");
    Serial.println("PASSWORD: " + passwordForWiFi);
}
String s = "<a href='/'> Go Back </a>";
server.send(200, "text/html", s);
}
void reconnect() {
    while (!client.connected()) {
        if (client.connect("batteryesp", mqttUser, mqttPassword))
        {
            client.publish("test", keyDevice);
            client.subscribe(keyDevice);
            client.subscribe("set/Status");
        } else {
            delay(5000);
        }
    }
}
void getStatus() {
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
    client.subscribe("set/Status");
    if (!client.connected()) {
        reconnect();
        client.publish("casa", keyDevice);
    }
    pinMode(RELAYPIN1, OUTPUT);
    pinMode(RELAYPIN2, OUTPUT);
}
void callback(char* topic, byte* payload, unsigned int length)
{
    Serial.print("Message arrived in topic: ");

```

```

Serial.println(topic);
Serial.print("Message:");
recvData = "";
for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
    recvData += (char)payload[i];
}
Serial.println();
Serial.println("-----");
if (recvData == "1"){
    Serial.println("Relay on");
    digitalWrite(RELAYPIN1, HIGH);
    digitalWrite(RELAYPIN2, HIGH);
}
if (recvData == "0"){
    Serial.println("Relay off");
    digitalWrite(RELAYPIN1, LOW);
    digitalWrite(RELAYPIN2, LOW);
}
}

void setup() {
    Serial.begin(9600);
    Serial.println("");
    WiFi.mode(WIFI_AP_STA);
    WiFi.softAP(ssid, password); //Start HOTspot removing
password will disable security
    IPAddress myIP = WiFi.softAPIP(); //Get IP address
    Serial.print("HotSpt IP:");
    Serial.println(myIP);
    server.on("/", handleRoot); //Which routine to handle at
root location
    server.on ("/action_page", handleSave);
    server.begin(); //Start server
    Serial.println("HTTP server started");
}

void loop() {

```

```
client.loop();
if (flag == false) {
    server.handleClient();
}
if (flag == true) {
    int numberOfNetworks = WiFi.scanNetworks();
    for (int i = 0; i < numberOfNetworks; i++) {
        if (loginForWiFi == WiFi.SSID(i)) {
            Serial.println(WiFi.SSID(i));
            WiFi.softAPdisconnect (true);
            WiFi.begin(loginForWiFi, passwordForWiFi);
            Serial.println(WiFi.waitForConnectResult());
            if (WiFi.waitForConnectResult() == WL_CONNECTED) {
                Serial.println("Connect good");
                getStatus();
            } else {
                Serial.println("Password faild");
                WiFi.softAP(ssid, password);
                flag = false;
            }
        } else {
            if (flag == true) {
                Serial.println("NOT found");
                flag = false;
            }
        }
    }
}
}
```